ИНСТИТУТ ИНТЕЛЛЕКТУАЛЬНЫХ КИБЕРНЕТИЧЕСКИХ СИСТЕМ

Кафедра
«Криптология и кибербезопасность»

Лабораторная работа №3

по предмету «Технологии контейнеризации»

Выполнил студент группы Б20-505

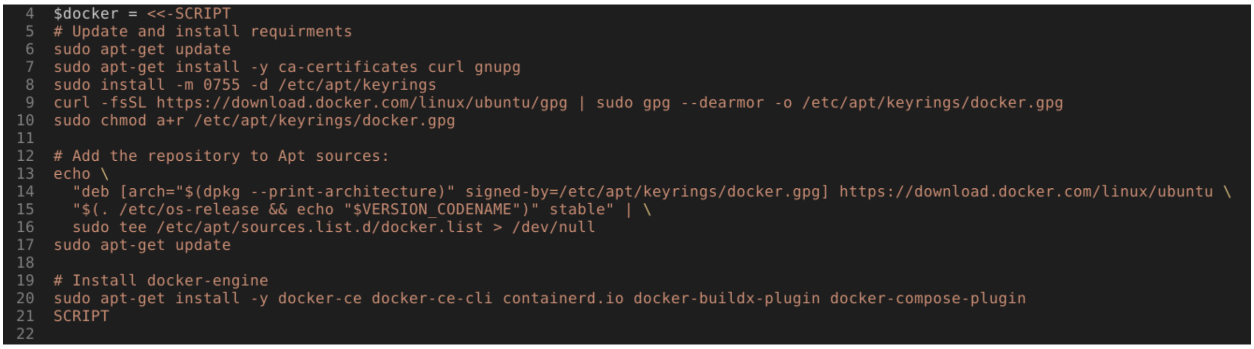
Сорочан Илья

Москва – 2023

Содержание

1. Установка docker

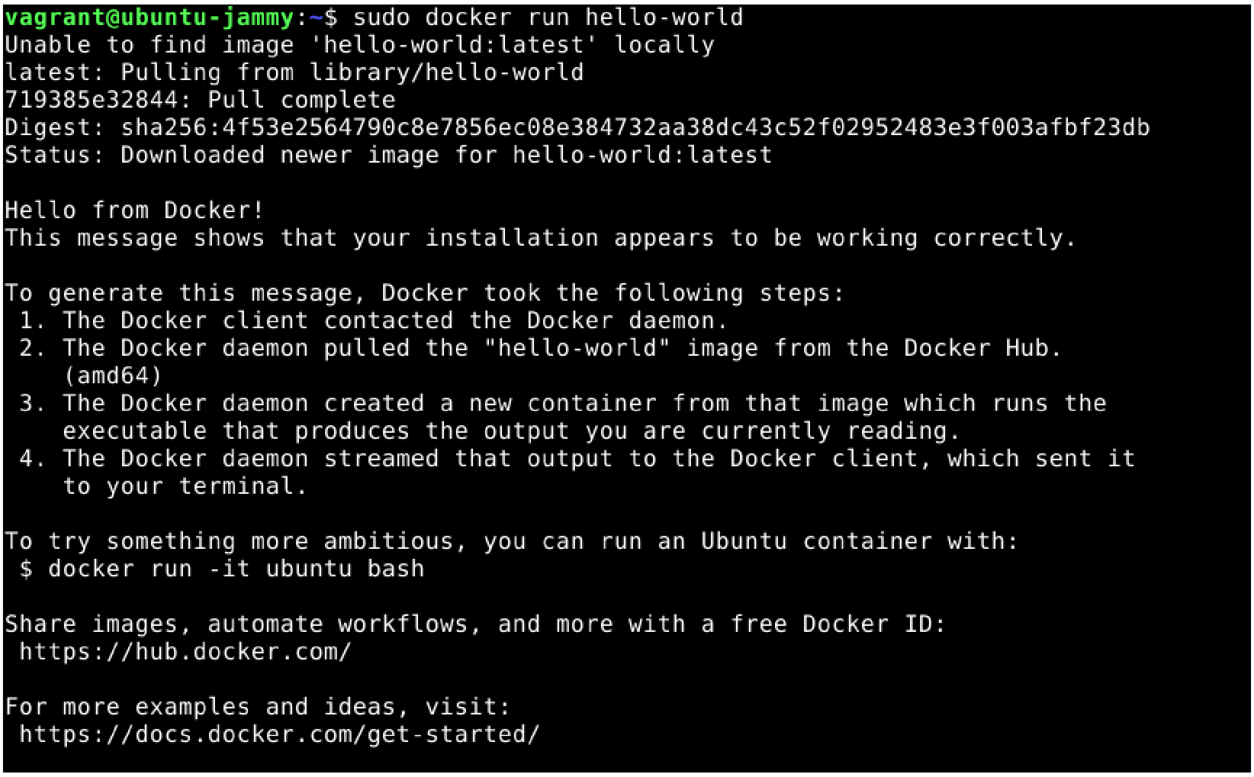
Для установки docker engine был использован скрипт из прошлых лабораторных работ (который запускался на этапе provision):



```
$docker = <<-SCRIPT
# Update and install requirments
sudo apt-get update
sudo apt-get install -y ca-certificates curl gnupg
sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
sudo chmod a+r /etc/apt/keyrings/docker.gpg

# Add the repository to Apt sources:
echo \
  "deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \
  "$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update

# Install docker-engine
sudo apt-get install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
SCRIPT

```

Рисунок 1: Скрипт для установки docker engine

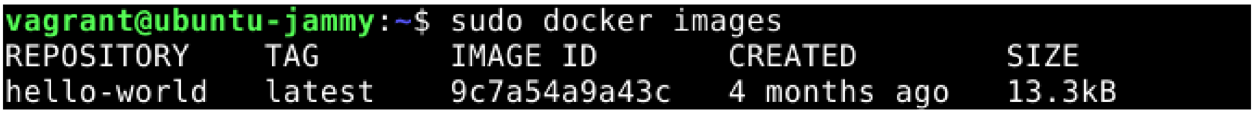
Проверка корректности установки:



```
vagrant@ubuntu-jammy:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
719385e32844: Pull complete
Digest: sha256:4f53e2564790c8e7856ec08e384732aa38dc43c52f02952483e3f003afbf23db
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

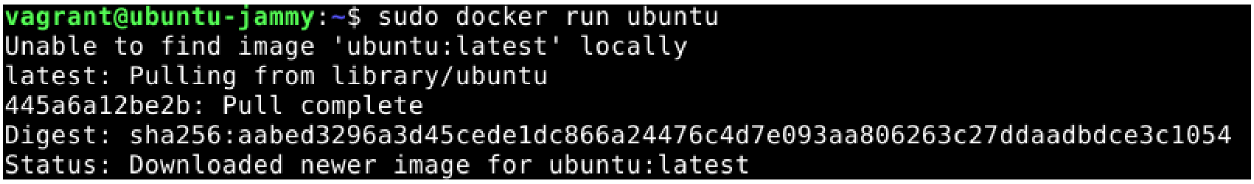
Рисунок 2: Запуск контейнера hello-world

2. Образы (images)



```
vagrant@ubuntu-jammy:~$ sudo docker images
REPOSITORY     TAG       IMAGE ID       CREATED         SIZE
hello-world    latest    9c7a54a9a43c   4 months ago    13.3kB
```

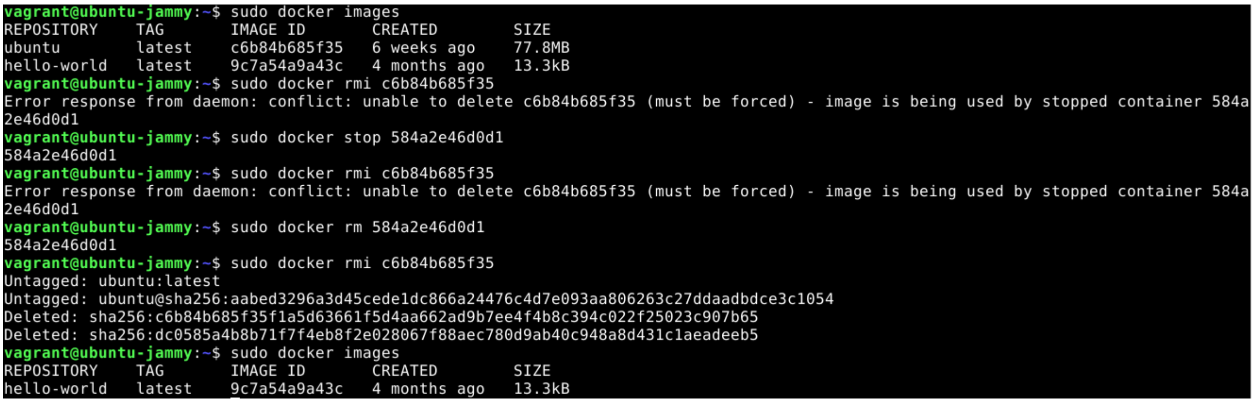
Рисунок 3: Список образов Docker



```
vagrant@ubuntu-jammy:~$ sudo docker run ubuntu
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
445a6a12be2b: Pull complete
Digest: sha256:aabed3296a3d45cede1dc866a24476c4d7e093aa806263c27ddaadbdce3c1054
Status: Downloaded newer image for ubuntu:latest
```

Рисунок 4: Загрузка образа Ubuntu

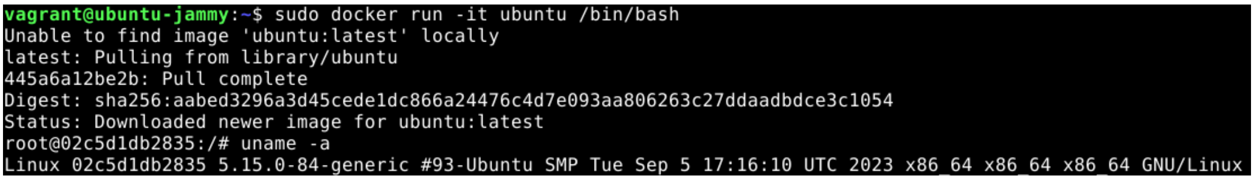
Для того что бы удалить образ необходимо сначала удалиить контейнер (на рисунке показана попытка удалить образ остановив контейнер, но не удалив его):



```
vagrant@ubuntu-jammy:~$ sudo docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
ubuntu        latest    c6b84b685f35   6 weeks ago    77.8MB
hello-world   latest    9c7a54a9a43c   4 months ago   13.3kB
vagrant@ubuntu-jammy:~$ sudo docker rmi c6b84b685f35
Error response from daemon: conflict: unable to delete c6b84b685f35 (must be forced) - image is being used by stopped container 584a
2e46d0d1
vagrant@ubuntu-jammy:~$ sudo docker stop 584a2e46d0d1
584a2e46d0d1
vagrant@ubuntu-jammy:~$ sudo docker rmi c6b84b685f35
Error response from daemon: conflict: unable to delete c6b84b685f35 (must be forced) - image is being used by stopped container 584a
2e46d0d1
vagrant@ubuntu-jammy:~$ sudo docker rm 584a2e46d0d1
584a2e46d0d1
vagrant@ubuntu-jammy:~$ sudo docker rmi c6b84b685f35
Untagged: ubuntu:latest
Untagged: ubuntu@sha256:aabed3296a3d45cede1dc866a24476c4d7e093aa806263c27ddaadbdce3c1054
Deleted: sha256:c6b84b685f35f1a5d63661f5d4aa662ad9b7ee4f4b8c394c022f25023c907b65
Deleted: sha256:dc0585a4b8b71f7f4eb8f2e028067f88aec780d9ab40c948a8d431c1aeadeeb5
vagrant@ubuntu-jammy:~$ sudo docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world   latest    9c7a54a9a43c   4 months ago   13.3kB
```

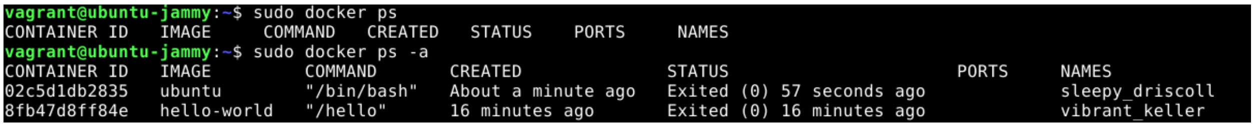
Рисунок 5: Удаление образа Ubuntu

3. Запуск контейнера



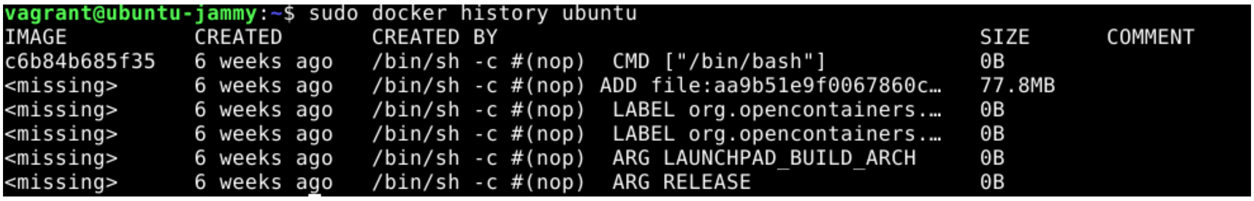
```
vagrant@ubuntu-jammy:~$ sudo docker run -it ubuntu /bin/bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
445a6a12be2b: Pull complete
Digest: sha256:aabed3296a3d45cede1dc866a24476c4d7e093aa806263c27ddaadbdce3c1054
Status: Downloaded newer image for ubuntu:latest
root@02c5d1db2835:/# uname -a
Linux 02c5d1db2835 5.15.0-84-generic #93-Ubuntu SMP Tue Sep 5 17:16:10 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
```

Рисунок 6: Запуск `/bin/bash` в интерактивном режиме



```
vagrant@ubuntu-jammy:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
vagrant@ubuntu-jammy:~$ sudo docker ps -a
CONTAINER ID   IMAGE         COMMAND       CREATED              STATUS                      PORTS     NAMES
02c5d1db2835   ubuntu        "/bin/bash"   About a minute ago   Exited (0) 57 seconds ago             sleepy_driscoll
8fb47d8ff84e   hello-world   "/hello"      16 minutes ago       Exited (0) 16 minutes ago             vibrant_keller
```

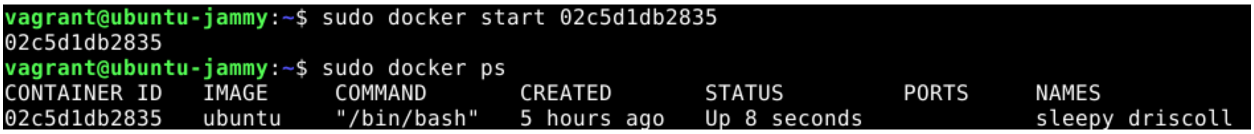
Рисунок 7: Списки запущенных и всех контейнеров



```
vagrant@ubuntu-jammy:~$ sudo docker history ubuntu
IMAGE          CREATED       CREATED BY                                      SIZE      COMMENT
c6b84b685f35   6 weeks ago   /bin/sh -c #(nop)  CMD ["/bin/bash"]            0B
<missing>      6 weeks ago   /bin/sh -c #(nop) ADD file:aa9b51e9f0067860c…   77.8MB
<missing>      6 weeks ago   /bin/sh -c #(nop)  LABEL org.opencontainers.…   0B
<missing>      6 weeks ago   /bin/sh -c #(nop)  LABEL org.opencontainers.…   0B
<missing>      6 weeks ago   /bin/sh -c #(nop)  ARG LAUNCHPAD_BUILD_ARCH     0B
<missing>      6 weeks ago   /bin/sh -c #(nop)  ARG RELEASE                  0B
```

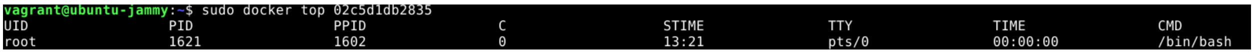
Рисунок 8: История контейнера ubuntu

4. Управление контейнерами



```
vagrant@ubuntu-jammy:~$ sudo docker start 02c5d1db2835
02c5d1db2835
vagrant@ubuntu-jammy:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND       CREATED       STATUS         PORTS     NAMES
02c5d1db2835   ubuntu    "/bin/bash"   5 hours ago   Up 8 seconds             sleepy_driscoll
```

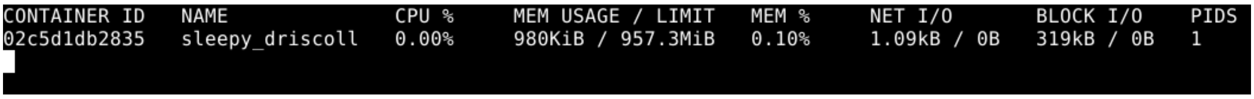
Рисунок 9: Запуск контейнера



```
vagrant@ubuntu-jammy:~$ sudo docker top 02c5d1db2835
UID                 PID                 PPID                C                   STIME               TTY                 TIME                CMD
root                1621                1602                0                   13:21               pts/0               00:00:00            /bin/bash
```

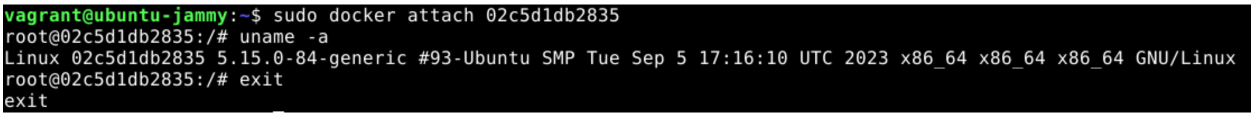
Рисунок 10: Просмотр запущенных процессов в контейнере

Команда `sudo docker stats 02c5d1db2835` выводит следующую информацию о контейнере:



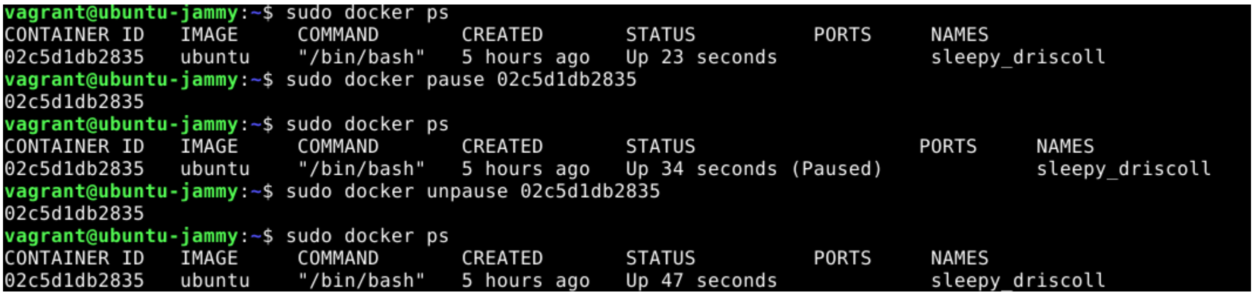
```
CONTAINER ID   NAME              CPU %     MEM USAGE / LIMIT   MEM %     NET I/O         BLOCK I/O     PIDS
02c5d1db2835   sleepy_driscoll   0.00%     980KiB / 957.3MiB   0.10%     1.09kB / 0B     319kB / 0B    1
```

Рисунок 11: Ресурсы, занимаемые контейнером



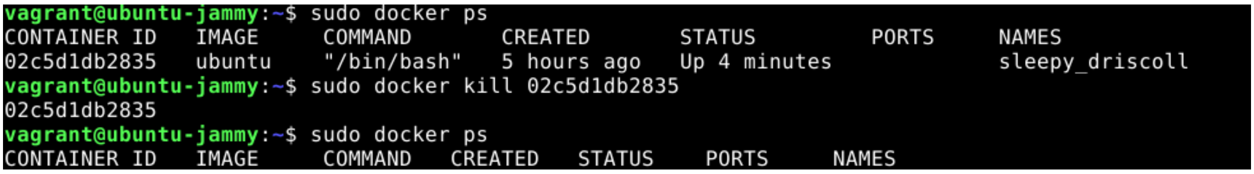
```
vagrant@ubuntu-jammy:~$ sudo docker attach 02c5d1db2835
root@02c5d1db2835:/# uname -a
Linux 02c5d1db2835 5.15.0-84-generic #93-Ubuntu SMP Tue Sep 5 17:16:10 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
root@02c5d1db2835:/# exit
exit
```

Рисунок 12: Подсоединение к контейнеру



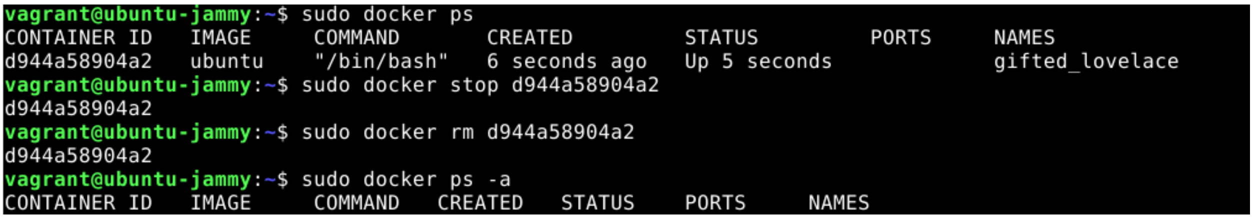
```
vagrant@ubuntu-jammy:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND       CREATED       STATUS          PORTS     NAMES
02c5d1db2835   ubuntu    "/bin/bash"   5 hours ago   Up 23 seconds             sleepy_driscoll
vagrant@ubuntu-jammy:~$ sudo docker pause 02c5d1db2835
02c5d1db2835
vagrant@ubuntu-jammy:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND       CREATED       STATUS                   PORTS     NAMES
02c5d1db2835   ubuntu    "/bin/bash"   5 hours ago   Up 34 seconds (Paused)             sleepy_driscoll
vagrant@ubuntu-jammy:~$ sudo docker unpause 02c5d1db2835
02c5d1db2835
vagrant@ubuntu-jammy:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND       CREATED       STATUS          PORTS     NAMES
02c5d1db2835   ubuntu    "/bin/bash"   5 hours ago   Up 47 seconds             sleepy_driscoll
```

Рисунок 13: Приостановка контейнера



```
vagrant@ubuntu-jammy:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND       CREATED       STATUS         PORTS     NAMES
02c5d1db2835   ubuntu    "/bin/bash"   5 hours ago   Up 4 minutes             sleepy_driscoll
vagrant@ubuntu-jammy:~$ sudo docker kill 02c5d1db2835
02c5d1db2835
vagrant@ubuntu-jammy:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

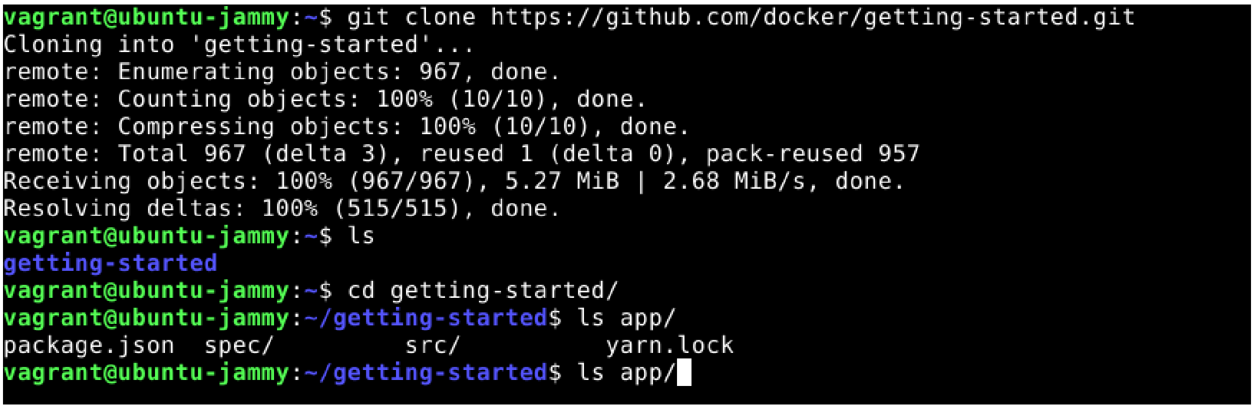
Рисунок 14: Удаление процессов контейнера



```
vagrant@ubuntu-jammy:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND       CREATED         STATUS         PORTS     NAMES
d944a58904a2   ubuntu    "/bin/bash"   6 seconds ago   Up 5 seconds             gifted_lovelace
vagrant@ubuntu-jammy:~$ sudo docker stop d944a58904a2
d944a58904a2
vagrant@ubuntu-jammy:~$ sudo docker rm d944a58904a2
d944a58904a2
vagrant@ubuntu-jammy:~$ sudo docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

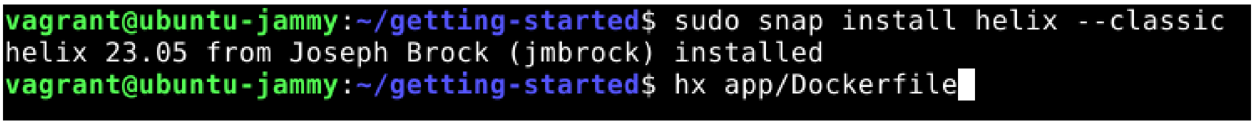
Рисунок 15: Остановка и удаление контейнера

5. Контейнеризация приложений



```
vagrant@ubuntu-jammy:~$ git clone https://github.com/docker/getting-started.git
Cloning into 'getting-started'...
remote: Enumerating objects: 967, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 967 (delta 3), reused 1 (delta 0), pack-reused 957
Receiving objects: 100% (967/967), 5.27 MiB | 2.68 MiB/s, done.
Resolving deltas: 100% (515/515), done.
vagrant@ubuntu-jammy:~$ ls
getting-started
vagrant@ubuntu-jammy:~$ cd getting-started/
vagrant@ubuntu-jammy:~/getting-started$ ls app/
package.json  spec/         src/          yarn.lock
vagrant@ubuntu-jammy:~/getting-started$ ls app/
```

Рисунок 16: Клонирование репозитория

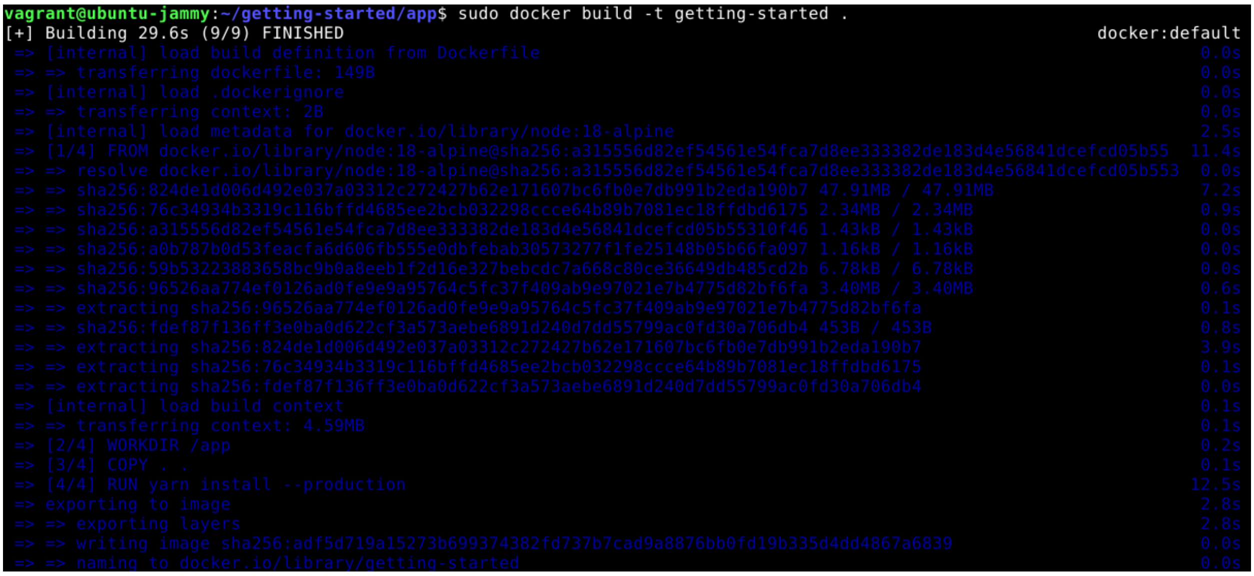


```
vagrant@ubuntu-jammy:~/getting-started$ sudo snap install helix --classic
helix 23.05 from Joseph Brock (jmbrock) installed
vagrant@ubuntu-jammy:~/getting-started$ hx app/Dockerfile
```

Рисунок 17: Создание Dockerfile внутри `gettning-started/app`

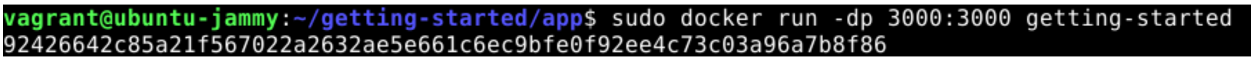
Внутри файла следующее содержимое:

```
FROM node:18-alpine
WORKDIR /app
COPY . .
RUN yarn install --production
CMD ["node", "src/index.js"]
EXPOSE 3000
```



```
vagrant@ubuntu-jammy:~/getting-started/app$ sudo docker build -t getting-started .
[+] Building 29.6s (9/9) FINISHED                                                                         docker:default
 => [internal] load build definition from Dockerfile                                                                0.0s
 => => transferring dockerfile: 149B                                                                                0.0s
 => [internal] load .dockerignore                                                                                   0.0s
 => => transferring context: 2B                                                                                     0.0s
 => [internal] load metadata for docker.io/library/node:18-alpine                                                   2.5s
 => [1/4] FROM docker.io/library/node:18-alpine@sha256:a315556d82ef54561e54fca7d8ee333382de183d4e56841dcefcd05b55  11.4s
 => => resolve docker.io/library/node:18-alpine@sha256:a315556d82ef54561e54fca7d8ee333382de183d4e56841dcefcd05b553 0.0s
 => => sha256:824de1d006d492e037a03312c272427b62e171607bc6fb0e7db991b2eda190b7 47.91MB / 47.91MB                    7.2s
 => => sha256:76c34934b3319c116bffd4685ee2bcb032298ccce64b89b7081ec18ffdbd6175 2.34MB / 2.34MB                      0.9s
 => => sha256:a315556d82ef54561e54fca7d8ee333382de183d4e56841dcefcd05b55310f46 1.43kB / 1.43kB                      0.0s
 => => sha256:a0b787b0d53feacfa6d606fb555e0dbfebab30573277f1fe25148b05b66fa097 1.16kB / 1.16kB                      0.0s
 => => sha256:59b53223883658bc9b0a8eeb1f2d16e327bebcdc7a668c80ce36649db485cd2b 6.78kB / 6.78kB                      0.0s
 => => sha256:96526aa774ef0126ad0fe9e9a95764c5fc37f409ab9e97021e7b4775d82bf6fa 3.40MB / 3.40MB                      0.6s
 => => extracting sha256:96526aa774ef0126ad0fe9e9a95764c5fc37f409ab9e97021e7b4775d82bf6fa                           0.1s
 => => sha256:fdef87f136ff3e0ba0d622cf3a573aebe6891d240d7dd55799ac0fd30a706db4 453B / 453B                          0.8s
 => => extracting sha256:824de1d006d492e037a03312c272427b62e171607bc6fb0e7db991b2eda190b7                           3.9s
 => => extracting sha256:76c34934b3319c116bffd4685ee2bcb032298ccce64b89b7081ec18ffdbd6175                           0.1s
 => => extracting sha256:fdef87f136ff3e0ba0d622cf3a573aebe6891d240d7dd55799ac0fd30a706db4                           0.0s
 => [internal] load build context                                                                                   0.1s
 => => transferring context: 4.59MB                                                                                 0.1s
 => [2/4] WORKDIR /app                                                                                              0.2s
 => [3/4] COPY . .                                                                                                  0.1s
 => [4/4] RUN yarn install --production                                                                            12.5s
 => exporting to image                                                                                              2.8s
 => => exporting layers                                                                                             2.8s
 => => writing image sha256:adf5d719a15273b699374382fd737b7cad9a8876bb0fd19b335d4dd4867a6839                        0.0s
 => => naming to docker.io/library/getting-started                                                                  0.0s
```

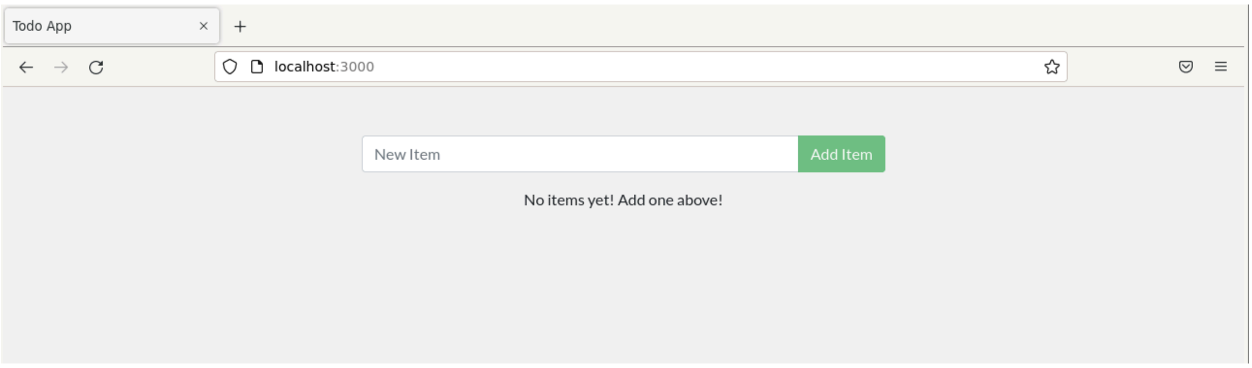
Рисунок 18: Сборка контейнера



```
vagrant@ubuntu-jammy:~/getting-started/app$ sudo docker run -dp 3000:3000 getting-started
92426642c85a21f567022a2632ae5e661c6ec9bfe0f92ee4c73c03a96a7b8f86
```

Рисунок 19: Запуск контейнера

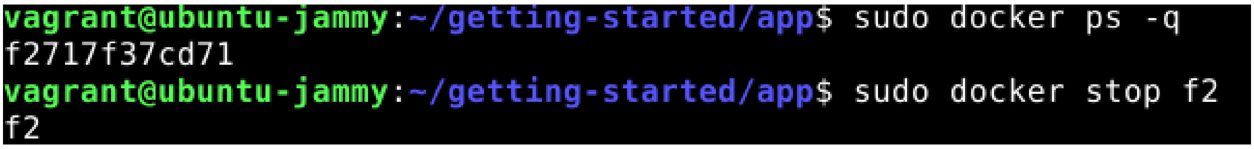
После добавление правила на проброс порта и перезапуска виртуальной машины (не забываем снова запустить контейнер) на `http://localhost:3000` высвечивается следующая страница:



No items yet! Add one above!

Рисунок 20: Проверка запущеного контейнера

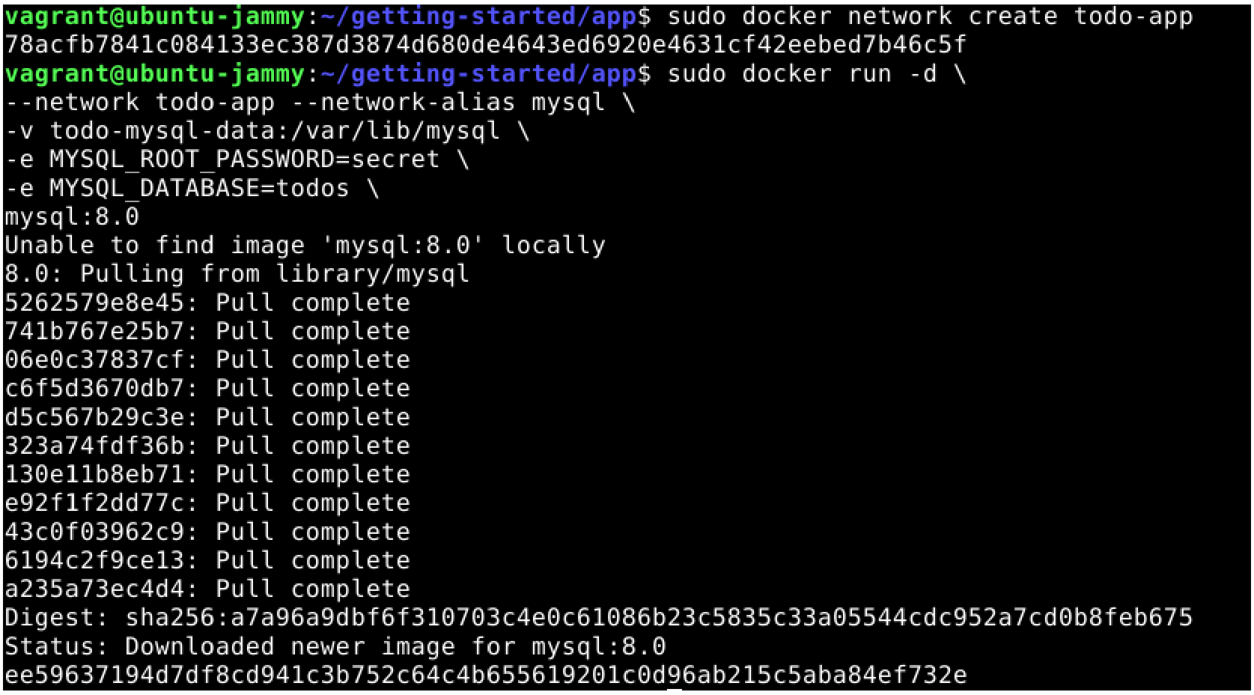
Остановить контейнер можно следующей командой:



```
vagrant@ubuntu-jammy:~/getting-started/app$ sudo docker ps -q
f2717f37cd71
vagrant@ubuntu-jammy:~/getting-started/app$ sudo docker stop f2
f2
```

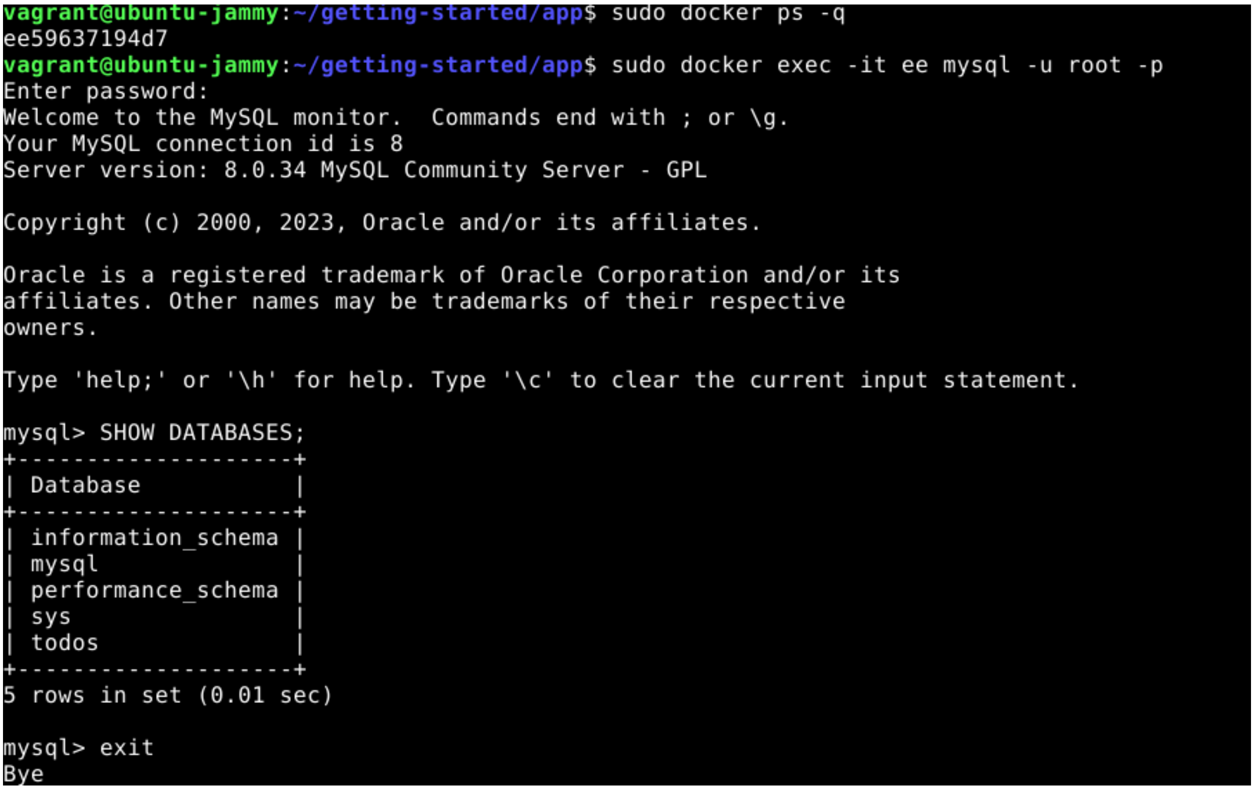
Рисунок 21: Остановка контейнера

6. Мультиконтейнерные приложения



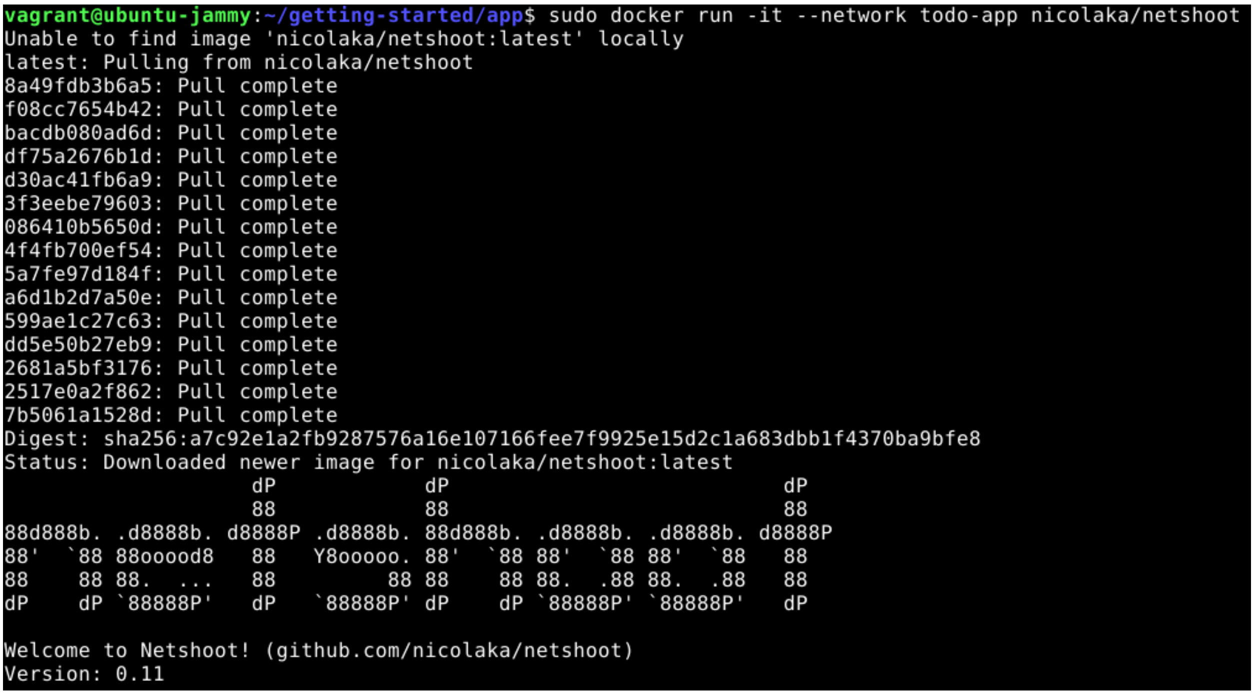
```
vagrant@ubuntu-jammy:~/getting-started/app$ sudo docker network create todo-app
78acfb7841c084133ec387d3874d680de4643ed6920e4631cf42eebed7b46c5f
vagrant@ubuntu-jammy:~/getting-started/app$ sudo docker run -d \
--network todo-app --network-alias mysql \
-v todo-mysql-data:/var/lib/mysql \
-e MYSQL_ROOT_PASSWORD=secret \
-e MYSQL_DATABASE=todos \
mysql:8.0
Unable to find image 'mysql:8.0' locally
8.0: Pulling from library/mysql
5262579e8e45: Pull complete
741b767e25b7: Pull complete
06e0c37837cf: Pull complete
c6f5d3670db7: Pull complete
d5c567b29c3e: Pull complete
323a74fdf36b: Pull complete
130e11b8eb71: Pull complete
e92f1f2dd77c: Pull complete
43c0f03962c9: Pull complete
6194c2f9ce13: Pull complete
a235a73ec4d4: Pull complete
Digest: sha256:a7a96a9dbf6f310703c4e0c61086b23c5835c33a05544cdc952a7cd0b8feb675
Status: Downloaded newer image for mysql:8.0
ee59637194d7df8cd941c3b752c64c4b655619201c0d96ab215c5aba84ef732e
```

Рисунок 22: Создание сети и поднятие mysql



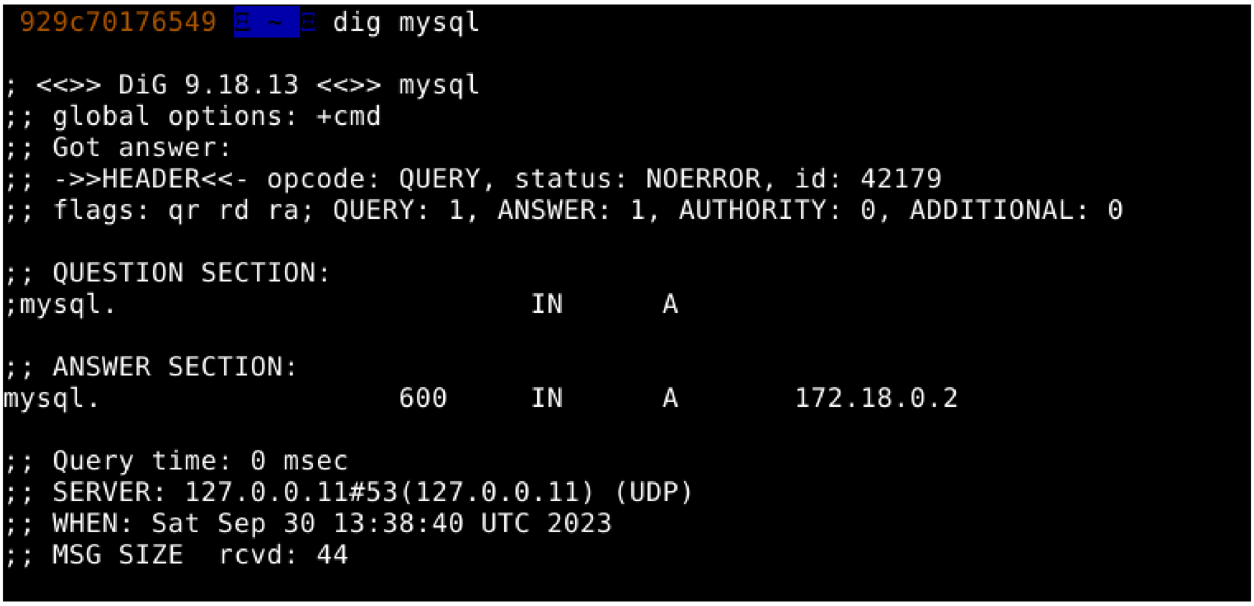
```
vagrant@ubuntu-jammy:~/getting-started/app$ sudo docker ps -q
ee59637194d7
vagrant@ubuntu-jammy:~/getting-started/app$ sudo docker exec -it ee mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.34 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| todos              |
+--------------------+
5 rows in set (0.01 sec)

mysql> exit
Bye
```

Рисунок 23: Проверка mysql



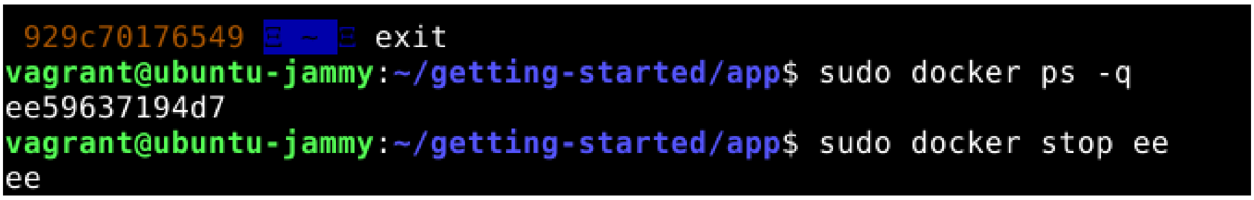
```
vagrant@ubuntu-jammy:~/getting-started/app$ sudo docker run -it --network todo-app nicolaka/netshoot
Unable to find image 'nicolaka/netshoot:latest' locally
latest: Pulling from nicolaka/netshoot
8a49fdb3b6a5: Pull complete
f08cc7654b42: Pull complete
bacdb080ad6d: Pull complete
df75a2676b1d: Pull complete
d30ac41fb6a9: Pull complete
3f3eebe79603: Pull complete
086410b5650d: Pull complete
4f4fb700ef54: Pull complete
5a7fe97d184f: Pull complete
a6d1b2d7a50e: Pull complete
599ae1c27c63: Pull complete
dd5e50b27eb9: Pull complete
2681a5bf3176: Pull complete
2517e0a2f862: Pull complete
7b5061a1528d: Pull complete
Digest: sha256:a7c92e1a2fb9287576a16e107166fee7f9925e15d2c1a683dbb1f4370ba9bfe8
Status: Downloaded newer image for nicolaka/netshoot:latest
                    dP            dP                           dP
                    88            88                           88
88d888b. .d8888b. d8888P .d8888b. 88d888b. .d8888b. .d8888b. d8888P
88'  `88 88ooood8   88   Y8ooooo. 88'  `88 88'  `88 88'  `88   88
88    88 88.  ...   88         88 88    88 88.  .88 88.  .88   88
dP    dP `88888P'   dP   `88888P' dP    dP `88888P' `88888P'   dP

Welcome to Netshoot! (github.com/nicolaka/netshoot)
Version: 0.11
```

Рисунок 24: Запуск контейнера `nicolaka/netshoot`



```
 929c70176549  ~  dig mysql

; <<>> DiG 9.18.13 <<>> mysql
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 42179
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;mysql.                         IN      A

;; ANSWER SECTION:
mysql.                  600     IN      A       172.18.0.2

;; Query time: 0 msec
;; SERVER: 127.0.0.11#53(127.0.0.11) (UDP)
;; WHEN: Sat Sep 30 13:38:40 UTC 2023
;; MSG SIZE  rcvd: 44
```

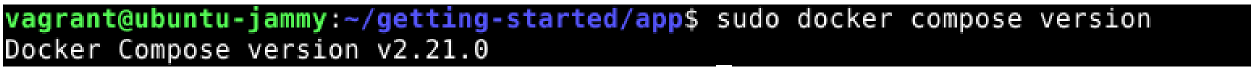
Рисунок 25: Результаты выполнения команды `dig mysql`



```
 929c70176549  ~  exit
vagrant@ubuntu-jammy:~/getting-started/app$ sudo docker ps -q
ee59637194d7
vagrant@ubuntu-jammy:~/getting-started/app$ sudo docker stop ee
ee
```

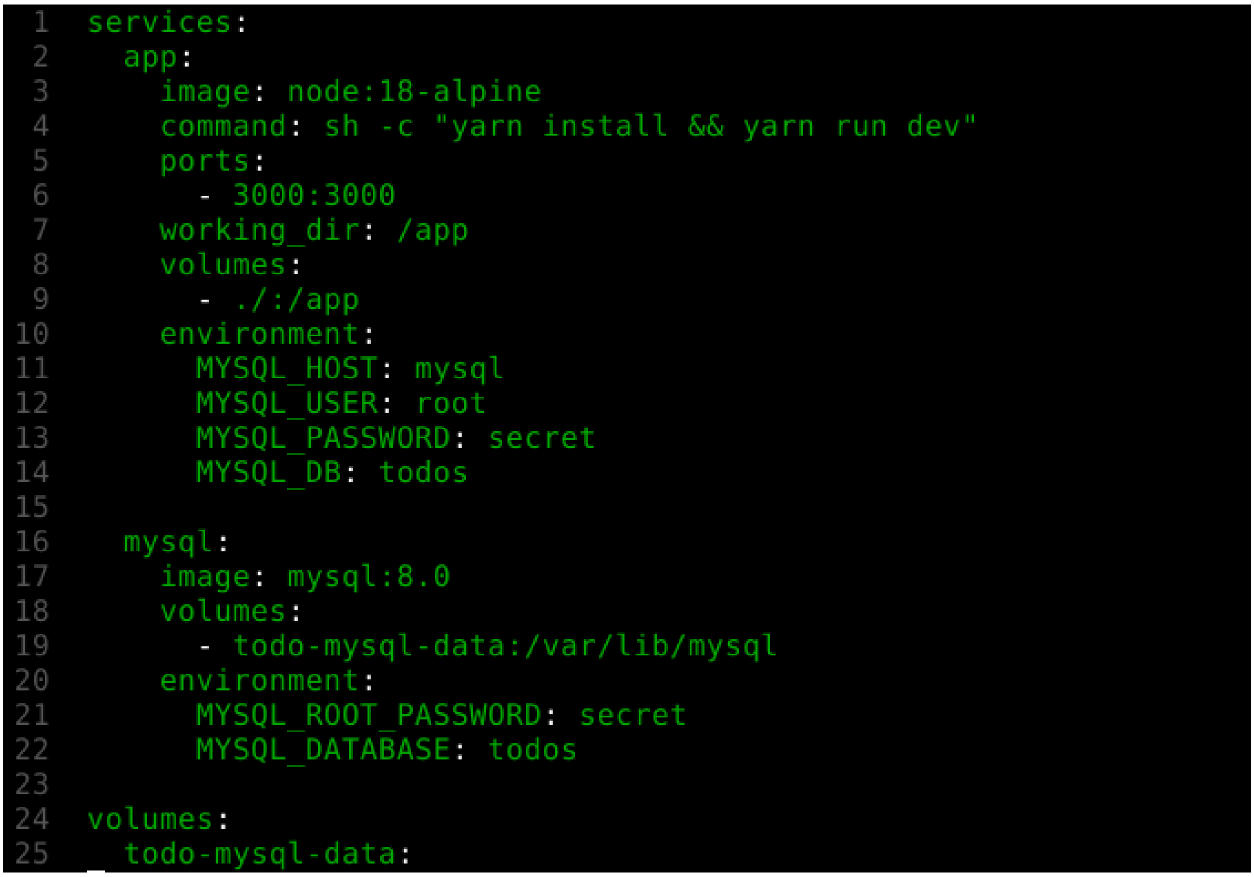
Рисунок 26: Завершение рабочих контейнеров

7. Docker compose



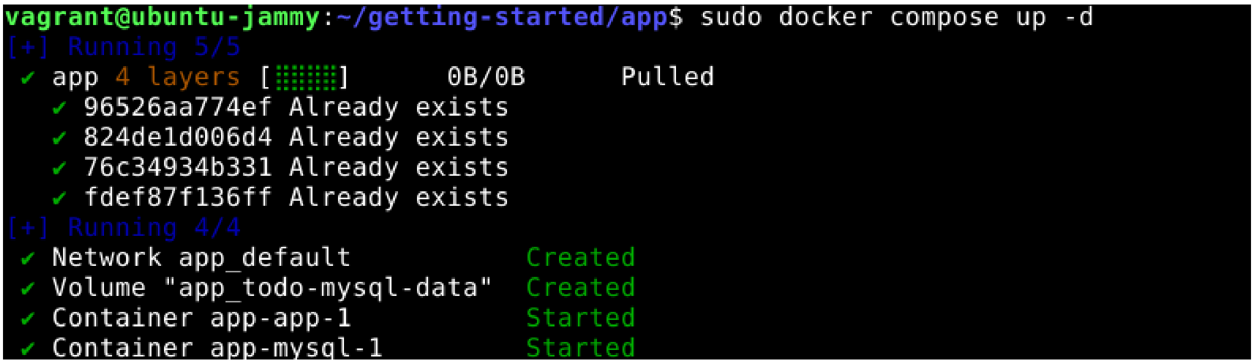
```
vagrant@ubuntu-jammy:~/getting-started/app$ sudo docker compose version
Docker Compose version v2.21.0
```

Рисунок 27: Версия docker compose



```
services:
  app:
    image: node:18-alpine
    command: sh -c "yarn install && yarn run dev"
    ports:
      - 3000:3000
    working_dir: /app
    volumes:
      - ./:/app
    environment:
      MYSQL_HOST: mysql
      MYSQL_USER: root
      MYSQL_PASSWORD: secret
      MYSQL_DB: todos

  mysql:
    image: mysql:8.0
    volumes:
      - todo-mysql-data:/var/lib/mysql
    environment:
      MYSQL_ROOT_PASSWORD: secret
      MYSQL_DATABASE: todos

volumes:
  todo-mysql-data:
```

Рисунок 28: Файл `docker-compose.yml`



```
vagrant@ubuntu-jammy:~/getting-started/app$ sudo docker compose up -d
[+] Running 5/5
 ✔ app 4 layers [⣿⣿⣿⣿]      0B/0B      Pulled
   ✔ 96526aa774ef Already exists
   ✔ 824de1d006d4 Already exists
   ✔ 76c34934b331 Already exists
   ✔ fdef87f136ff Already exists
[+] Running 4/4
 ✔ Network app_default            Created
 ✔ Volume "app_todo-mysql-data"   Created
 ✔ Container app-app-1            Started
 ✔ Container app-mysql-1          Started
```

Рисунок 29: Запуск контейнеров при помощи docker compose